(12) UK Patent Application (19) GB (11) 2 347 833 (13) A

(43) Date of A Publication 13.09.2000

(21) Application No 9905482.7

(22) Date of Filing 11.03.1999

(71) Applicant(s)
3Com Technologies
(Incorporated in the Cayman Islands)
PO Box 309, Upland House, Georgetown, Grand Cayman, Cayman Islands

(72) Inventor(s)
Vincent Gavin
Con Cremin
Christopher Gilbert
Tadhg Creedon

(74) Agent and/or Address for Service
Bowles Horton
Felden House, Dower Mews, High Street, BERKHAMSTED, Herts, HP4 2BL, United Kingdom

(51) INT CL[7]
H04L 12/56

(52) UK CL (Edition R)
H4P PPS
H4K KTK

(56) Documents Cited
EP 0772326 A1

(58) Field of Search
UK CL (Edition Q) H4K KTK, H4P PPS
INT CL[6] H04L 12/56
Online Databases: WPI, EPODOC, JAPIO

(54) Abstract Title
Initiating flow control over packet data links

(57) A network interface for a packet-based communication system receives packets over a link 10 and directs them to a multiplicity of queues 15 on the basis of traffic type and/or priority. Flow control is initiated on the link to inhibit the sending of packets to the interface when a queue reaches a predetermined fill level. For each of the queues there are means for selectively enabling and disabling the initiation of flow control in response to a defined length of the respective queue, so that flow control can be initiated for some types of traffic and not for others. The queues may be established in FIFOs with the aid of write and read pointers and the separation of the pointers in terms of memory space may be used to govern the onset and cessation of flow control.

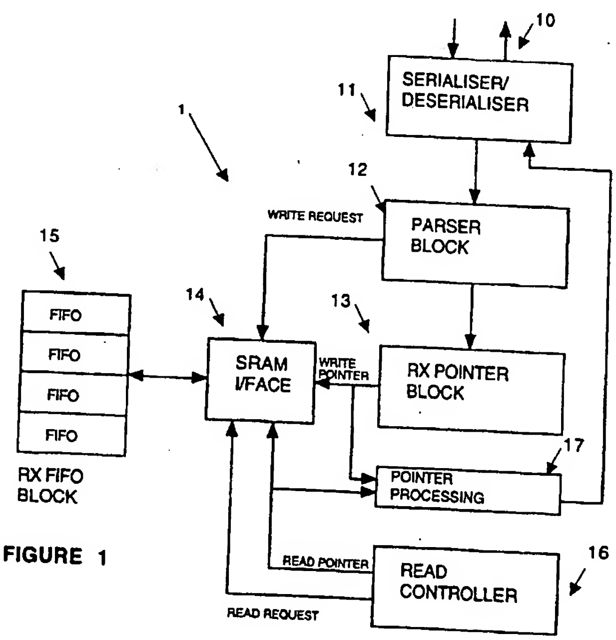


FIGURE 1

GB 2 347 833 A

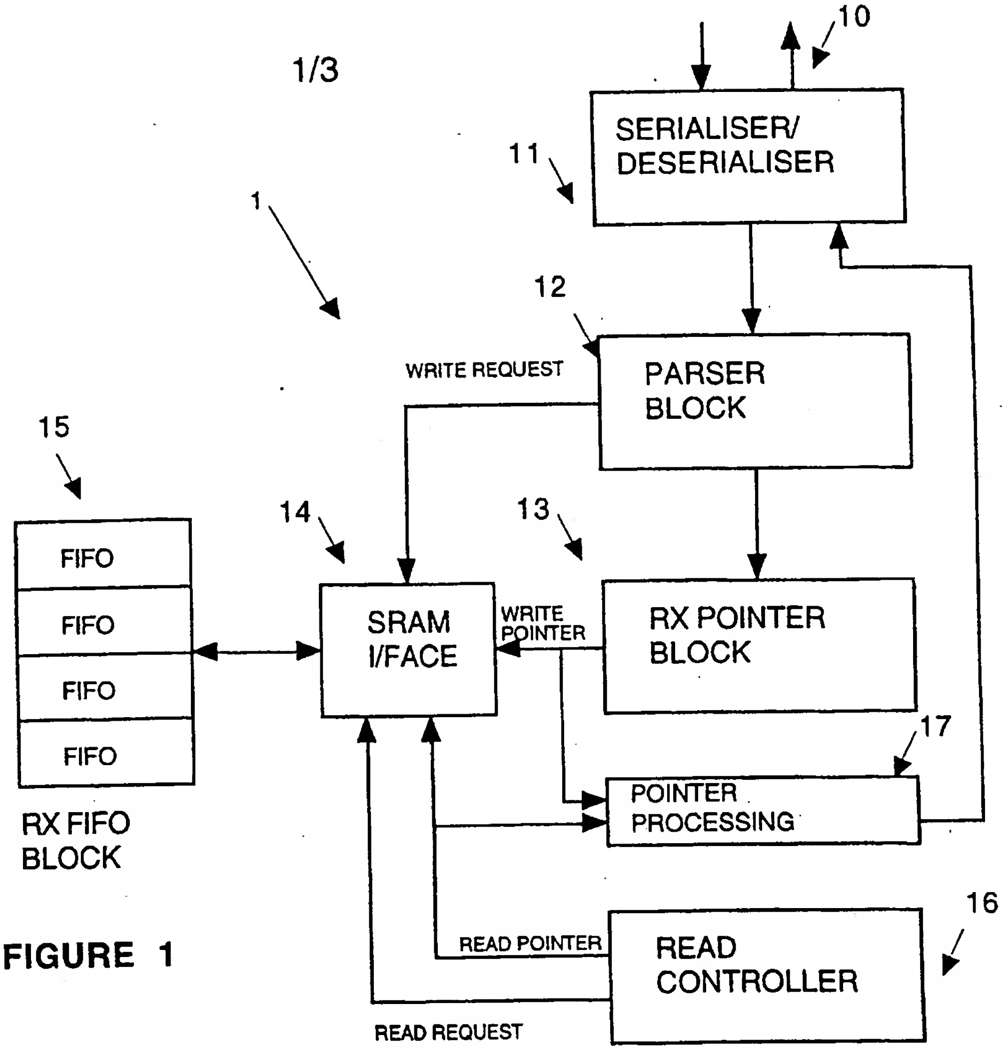


FIGURE 1

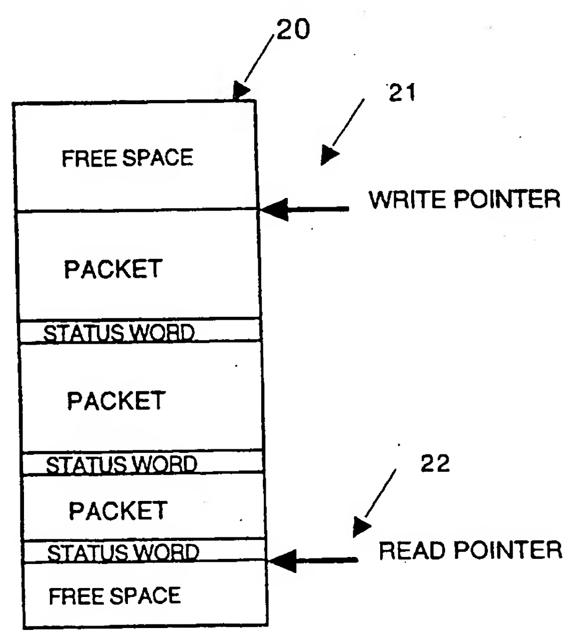


FIGURE 2

FIGURE 3

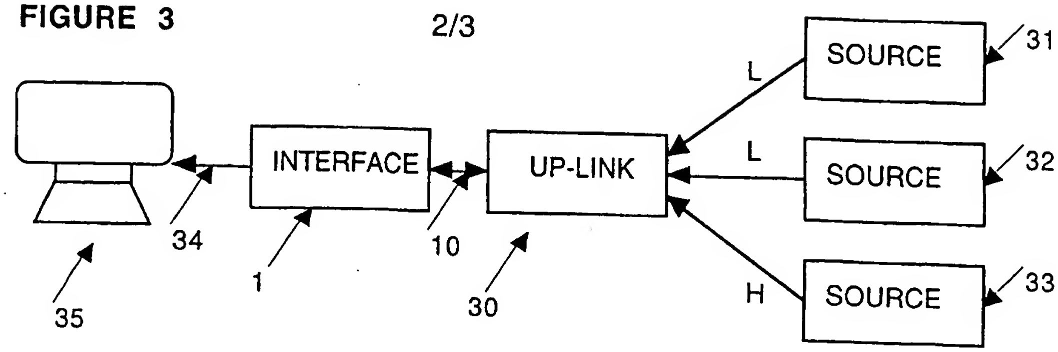


FIGURE 4

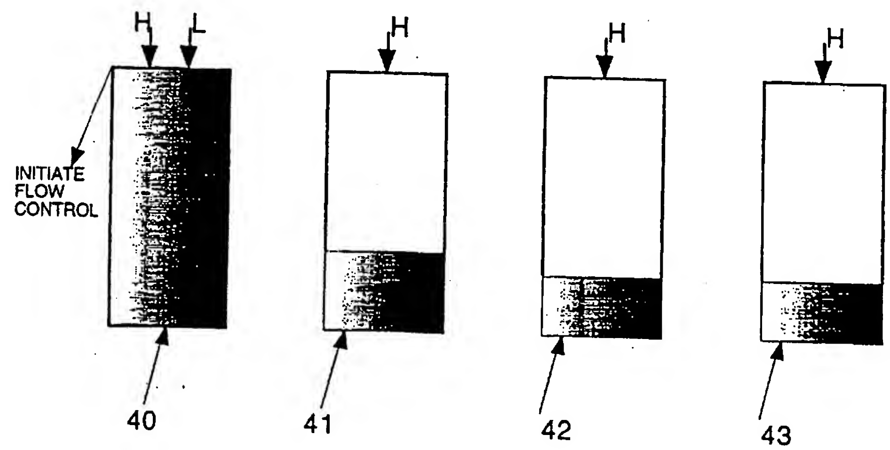


FIGURE 5

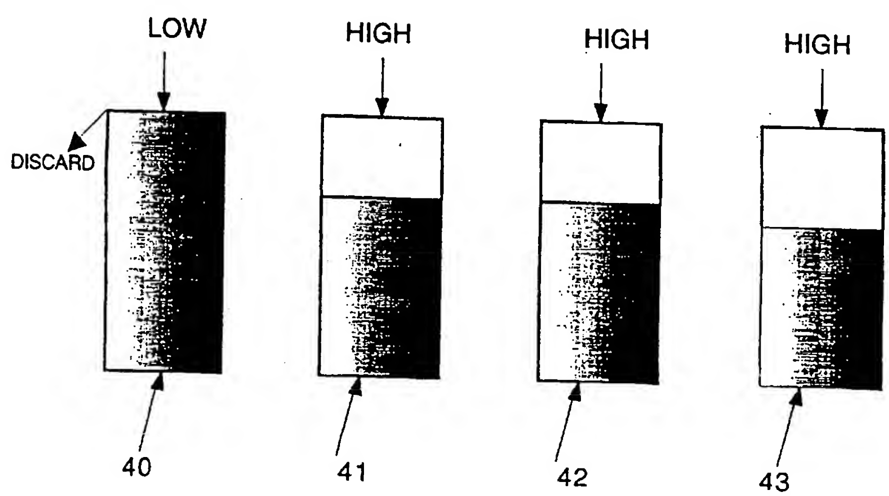
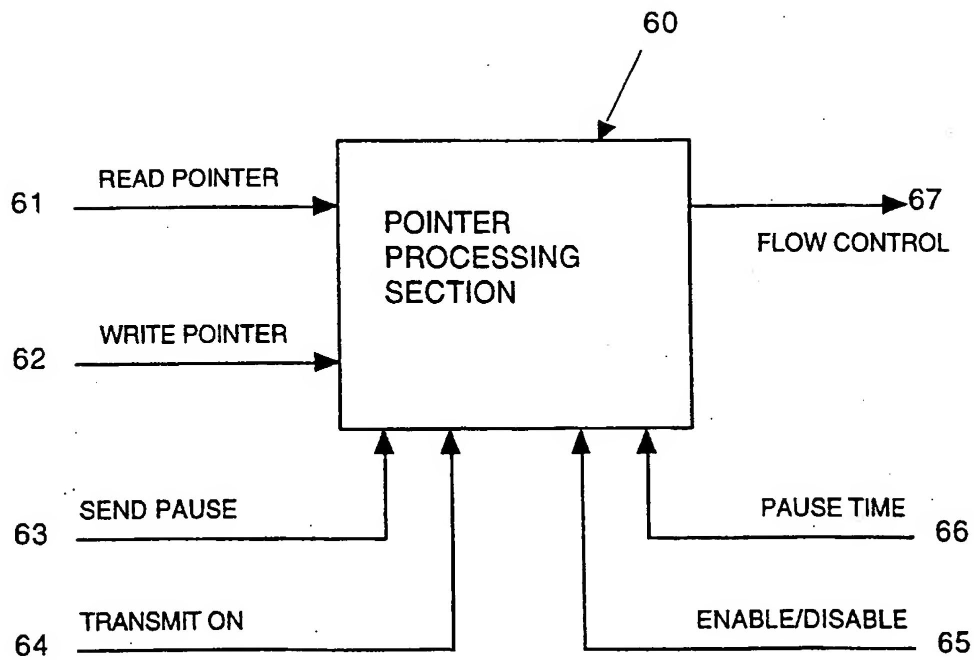


FIGURE 6


60
READ POINTER
61
POINTER
PROCESSING
SECTION
67
FLOW CONTROL
WRITE POINTER
62
SEND PAUSE
63
PAUSE TIME
66
TRANSMIT ON
64
ENABLE/DISABLE
65

METHOD AND APPARATUS FOR INITIATING FLOW CONTROL OVER PACKET DATA LINKS

Field of the Invention

This invention relates generally to packet-based data networks such as Ethernet networks, wherein data is conveyed in individual packets generally in the form of a header, address data, control or other identifying data, and message data, which is customarily followed by cyclic redundancy code data. In particular, the invention relates to network devices or switches which receive data over a relatively high speed link, which may be a serial link operating at a data rate in excess of one gigabit per second, and distributes packets, after deserialisation as required, and appropriate parsing and look-ups, to a multiplicity of packet queues. Such packet queues may be formed in buffer memory and may take the form of defined FIFO stores, constituted by a predefined storage space in, for example, static random access memory, defined by pointers including a tail, or read, pointer delimiting the emptying process, and a head or write pointer delimiting the filling process. Such pointers are generally controlled by means of a CPU or switching ASIC.

Background to the Invention

The distribution of input packets to a multiplicity of packet queues, either by way of explicit ports or from a deserialiser by way of an appropriate interface, is normally necessary because packets will normally have to be transmitted over links, all other data paths having a lower data rate than the input link or channel. Furthermore, they may require to be delayed for an indeterminate time on the count of bandwidth restriction or congestion on a path to a respective destination.

It is known to place incoming traffic to a switch into the separate queues based on traffic type, such as high versus low priority, or multimedia versus data traffic, management control frames versus normal data and so on. The type of the traffic is indicated by relevant fields of control data within packets or frames and it is known to sort traffic on the basis of such data.

It is known to discard packets when a temporary store for those packets is full (discard on receive) or is unable to forward packets owing to conditions downstream (discard on transmit).

It is customary in various networks, and prescribed according to a network standard such as IEEE 802.3, to generate control frames or packets at a device to inhibit the sending of further packets from a source supplying that device. This process is known as applying back pressure or, specifically, 'flow control'. It may be employed automatically when a device reaches some defined state of congestion, such as having its respective receive queues (or any one of them) in a 'full' state, which may be defined as some appropriate proportion of the actual physical capacity of the relevant storage space.

A problem with this approach, as is recognised in the industry, is that when receive queues approach fullness, however that is defined, it is necessary to initiate flow control on the incoming link blindly, that is to say indiscriminately, so that all traffic is stopped, not merely the traffic for the relevant queue. The present invention is concerned with a practical solution of improving performance beyond that associated with such indiscriminate flow control.

Summary of the Invention

The basis of the invention is to provide flow control of the link on a per-queue basis. If for example it was decided that it was acceptable to drop multimedia traffic but not, for example, SNMP management frames, flow control would not be enabled for the former, but would be for the latter. In this example, a link would be paused only when the management queue were full.

It should be understood that the process of flow control is essentially one used over full duplex links in which the sender of the packets causing congestion in a device will, on receipt of flow control packets or frames, cease sending packets to the originator of the flow control frames but will resend those packets after some pause time which is specified in the flow control frames. Resending of packets may also be allowed by the sending of a special control frame.

In addition, it is feasible to provide different pause criteria for each queue. For example, one may have a low pause value for multimedia traffic to minimize latencies and a high pause value for data traffic to allow time to empty large queues.

These and other features of the invention will become apparent in the following description with reference to the accompanying Figures.

Brief Description of the Drawings

Figure 1 is a schematic diagram of part of a network interface;

Figure 2 is an explanatory diagram of a FIFO store;

Figure 3 is a diagram illustrating part of a network in which the invention may be employed;

Figure 4 is a schematic illustration of various FIFO stores in different states of fullness;

Figure 5 is another illustration of FIFO stores in various states of fullness; and

Figure 6 is schematic illustration of a pointer processing section forming part of a pointer processor in Figure 1.

Detailed Description of the Preferred Example

The present invention will now be described by way of example only with reference to an interface which is intended to receive data over a duplex link conveying data in serial form at a high rate, such as in excess of one gigabit per second, and which allocates data received over that link to a multiplicity of receive queues. The onward transmission or processing of packets received in those queues is not of importance to the invention. The interface may form part of a multi-port device which switches received packets received over that link and over other links

of lesser data rate to various ports in accordance with the addressing of received packets or may supply a time slotted data bus and so on according to preference.

The interface 1 shown in Figure 1 is configured for the reception of serial data over a duplex link 10 which is coupled to a serialiser/deserialiser 11. The serialiser/deserialiser converts serial input data into 10-bit wide words according to the well known 8-bit-10-bit coding. Packets composed of a multiplicity of such words pass through a parser block 12 which examines as necessary header and control data in each packet so as to obtain, in a manner known per se, a queue number for each individual packet on the basis of the type of the packet. The queue number is passed to an RX pointer block 13 which develops for the respective queue a 'write pointer' which is used by an SRAM interface 14 to write, at a time determined by a 'write request' from the parser block 12, the relevant packet into a respective FIFO defined in an RX FIFO block 15. The organisation of the FIFOs is explained below with reference to Figure 2. The interface 14 also receives from a read controller 16 a 'read request' and a 'read pointer' for controlling read-out from the FIFO's defined within RX FIFO block 15.

As thus far described the interface system described is of known form. In accordance with the invention Figure 1 includes a pointer processing block 17 which responds to the generated pointers and controls the generation of control frames which are transmitted back along link 10 by the serialiser/deserialiser 11, as will be described in the following.

Figure 2 illustrates one of the FIFOs defined in the receive FIFO block 15. It is constituted by a fixed amount of static random access memory and between the locations defined by a write pointer 21 and a read pointer 22 comprises packet data separated by status words which each specify (among other things) the length of the packet. When the write pointer reaches the 'physical' boundary of the FIFO it wraps around to the start and continues to place packets into the FIFO. When the write pointer comes within one address behind the read pointer the FIFO is full; when the read pointer catches up with the write pointer the FIFO is empty. However, for the purpose of generating flow control signals and otherwise, one normally defines fullness to be represented by some predetermined distance, less than the maximum, between the two pointers. The reason for this is an inherent delay between the time a packet is dispatched from

the far end of the link 10 to the interface, the processing of that packet, the generation of a flow control frame and the time for that flow control frame to reach the device at the far end of the link. In practice therefore this system, in common with known systems, may define fullness as a selected fraction of the maximum space within the FIFO.

Figure 3 illustrates a typical, though greatly simplified, situation in which the invention has relevance. The interface or switch 1 is coupled by means of the link 10 to an up-link 30 which receives packets from various sources 31, 32 and 33. In general there may be many more sources coupled to the up-link 30. Further, the interface 1 provides received data packets over a link 34 to a user terminal 35.

Typically, the sources of data packets arriving at device 1 may have a low priority or high priority according to the type. This is indicated in Figure 3 by the allocation of L (Low) to sources 31 and 32, and the designation H (High) to the source 33. The packet type is (in known manner) indicated within the respective packets.

In common practice, packets which are received over a link 10 and directed to a multiplicity of packet queues achieves distribution on the application of a hashing algorithm applied, for example, either to the source address or to the source address/destination address combination of incoming packets. Since there are many fewer packet queues than, in general, source addresses or source address/destination address combinations of packets input to the device, it is common to find that, as shown in Figure 4, a traffic queue 40 containing both high priority and low priority data becomes full (however this is defined) while other traffic queues 41 - 43 remain partly empty. It is in general the case that high priority traffic is infrequent and of low volume whereas low priority traffic tends to be of much heavier volume. Figure 4 specifically shows a situation in which one FIFO buffer is full with both high and low priority traffic whereas the other buffers are comparatively empty and contain only high priority traffic. The queue containing both high and low priority traffic tends to fill up quickly so that the link is paused, by the generation of the relevant MAC control frame on the link 10 back to device 30. The scheme is inefficient, because all traffic is stopped notwithstanding the availability of space within the buffers.

Figure 5 illustrates the same four FIFO buffers wherein the traffic has been separated on the basis of traffic type and particularly priority.

The significance of Figure 5 is as follows. It is possible to initiate flow control in response to the state of fullness of any of the FIFO buffers. By inhibiting the initiation of flow control, any of the buffers can be allowed to proceed to fullness and then, despite the fullness of that buffer, flow control will not be initiated, thereby allowing traffic to continue. Since the buffer whose initiation of flow control is inhibited may become full, it is necessary to discard packets destined for this buffer.

In order to obtain optional flow control initiated by any of the queues, only some simple hardware additions are required. These are the pointer processing sections of which one is shown in Figure 6. there is one of these sections in block 17 (Figure 1) for each of the traffic queues.

The pointer processing section 60 shown in Figure 6 receives on lines 61 and 62 signals identifying or representing the read and write pointers for the respective traffic queue. The section 60 computes the distance between the pointers in terms of data spaces between them. The section also receives on line 63 a signal denoted 'send pause' which determines the distance between the pointers required to initiate flow control. The section receives on line 64 a second signal, denoted 'transmit on', which defines the distance required between the pointers to cause cessation of flow control. It may for example be desirable to allow the buffer to become more empty before flow control ceases than the state of fullness when flow control is initiated.

The section 60 also receives on line 65 an enable/disable signal which determines whether flow control will be initiated at all. This is a selectable signal which may be manually or automatically selected depending on the traffic type. The pointer processing section also receives on line 66 a 'pause time' signal which determines the pause time that will be conveyed by a flow control frame if it is sent. This is also a signal which may be selected by an administrator or computed automatically from traffic statistics.

The pointer processing section is only required to initiate and stop flow control according to

the distance between the pointers and the limits prescribed by the pause and send signals. The MAC control frames will be output on a line 67 which may be coupled to the serialiser/deserialiser 11 for transmission back down the link 10.

The present arrangement is quite versatile. Traffic may be sorted according to a variety of criteria and the generation of flow control frames can be inhibited unless a queue of packets of one particular type (or a selected plurality of types) reaches fullness. The scheme requires or may require the discarding of packets from queues which become full yet have not been selected for the initiation of flow control frames.

CLAIMS

1. A network interface for a packet-based communication system comprising means for receiving over a link packets and directing said packets to a multiplicity of queues and means for initiating flow control on said link to inhibit the sending of packets to the interface, wherein the said means for initiating flow control comprises for each of said queues means for selectively enabling and disabling the initiation of flow control in response to a defined length of the respective queue.

2. An interface according to claim 1 wherein the means for initiating flow control includes means for selectively defining the queue lengths for which flow control will be initiated and stopped.

3. An interface according to claim 1 wherein the flow control is governed by control frames which define a pause time and the means for initiating flow control includes means for adjusting said pause time for control frames generated in response to the initiating of flow control by the respective queue.

4. An interface according to any foregoing claim wherein said means for receiving said packets sorts the packets into the queues on the basis of traffic type

5. A method of operating flow control which inhibits the sending of data packets to an interface in which received packets are sorted into a multiplicity of queues on the basis of traffic type, said method comprising:

(a) providing an enabling and disabling control for each of the queues;

(b) initiating flow control when a queue reaches a prescribed limit and when the respective control for that queue is enabling; and

(c) preventing the initiation of flow control in response to a queue when the respective control is disabling.

6. A method according to claim 5 wherein packets intended for a particular queue are discarded when the initiation of flow control by that queue is disabled and the queue becomes full.

The Patent Office

10

INVESTOR IN PEOPLE

Application No: GB 9905482.7
Claims searched: All

Examiner: Gareth Griffiths
Date of search: 18 August 1999

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.Q): H4K (KTK), H4P (PPS)

Int Cl (Ed.6): H04L 12/56

Other: Online Databases: WPI, EPODOC, JAPIO

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
A	EP0772326 A1 (SUN MICROSYSTEMS)	

X Document indicating lack of novelty or inventive step	A Document indicating technological background and/or state of the art.
Y Document indicating lack of inventive step if combined with one or more other documents of same category.	P Document published on or after the declared priority date but before the filing date of this invention.
& Member of the same patent family	E Patent document published on or after, but with priority date earlier than, the filing date of this application.